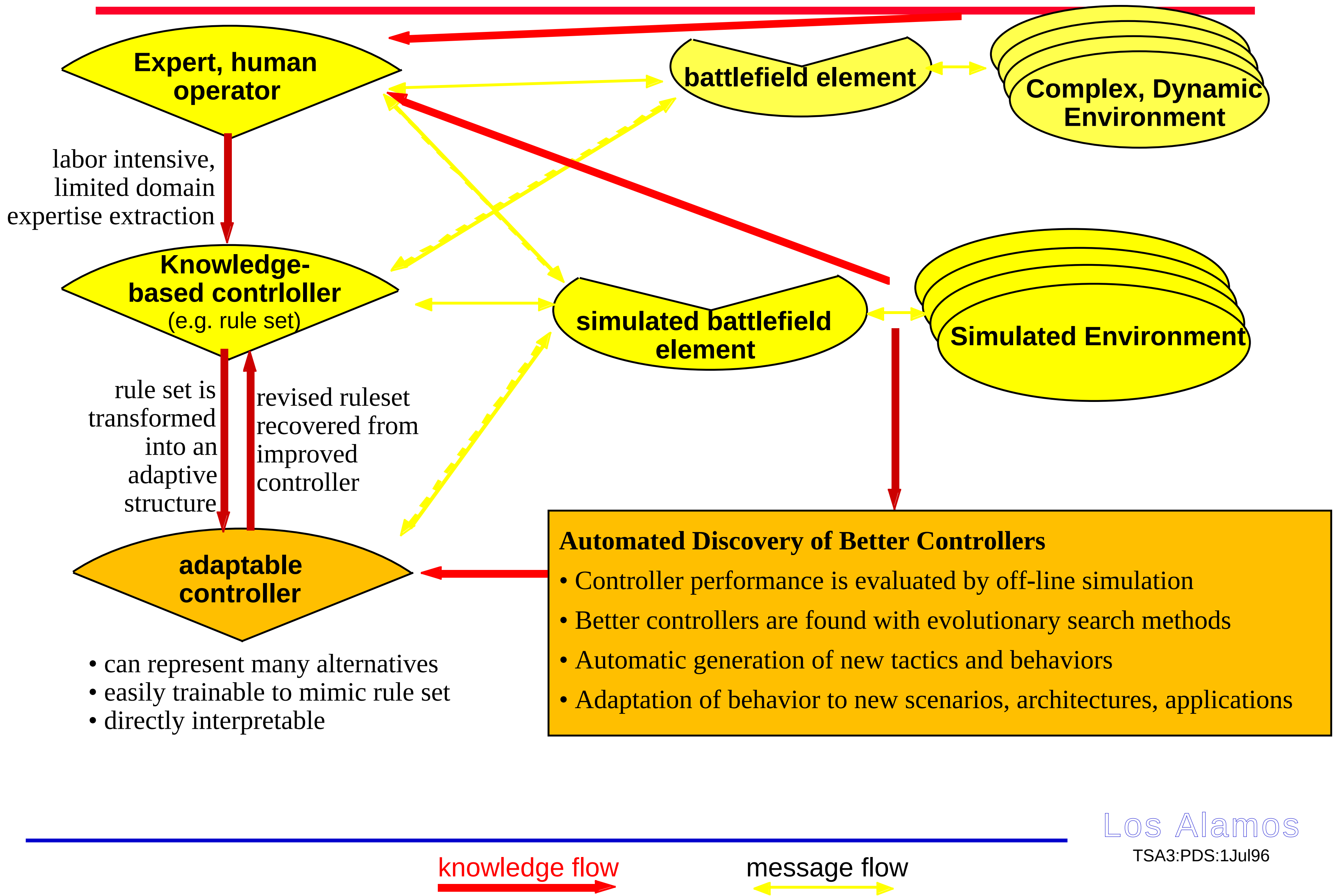
Automatic Simulation-based Learning in Knowledge-based Controllers

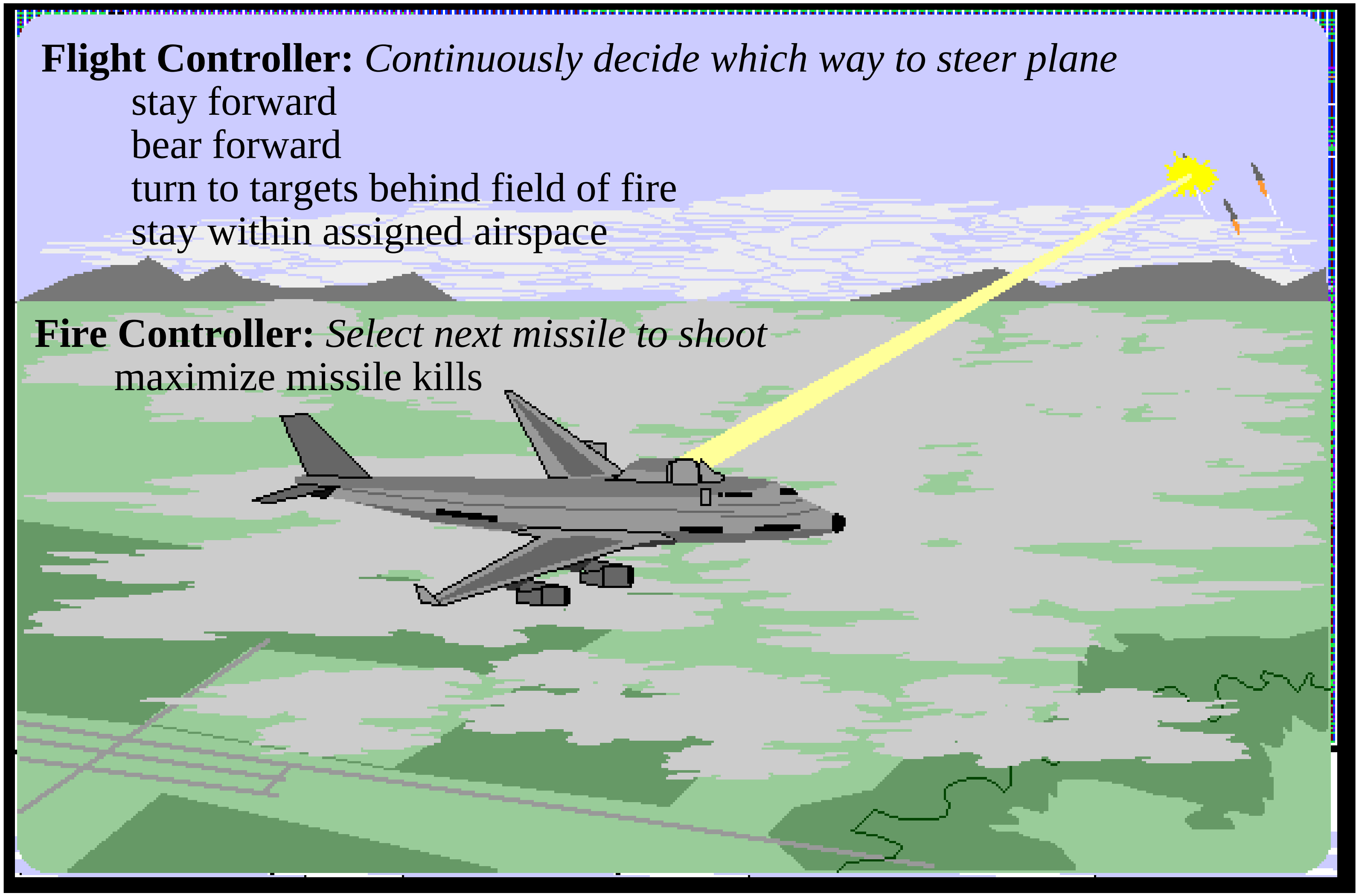
Phillip D. Stroud
Los Alamos National Laboratory
DMSO Individual Combatant Workshop
July 1-2, 1996

- Develop the Decision Making Component of Simulation Actors
  - discovery of new tactics and behavior
  - automated adaptation to new scenarios
  - intelligent interaction with complex, dynamic environments

Approach: convert rule set to adaptable controller, evolve better controller using simulation for performance evaluation



ABL Flight and Fire Controllers Provide Excellent Testbed to Investigate Intelligent Controllers



A Controller's Performance is Evaluated by Simulation

ELASTIC simulation package

Evolutionary Los Alamos Simulation-based Training for Intelligent Controllers

- ABL and TBM actors

  (+ fighters, TEL's, AWACS, ASM, AAM, Patriot, BMC3I in Samson, JointSim)

- High fidelity missile flyout and laser propagation
- Plug-in ABL controller modules
- Performance criteria: fraction of missiles killed
- Statistical significance requires thousands of simulated engagements

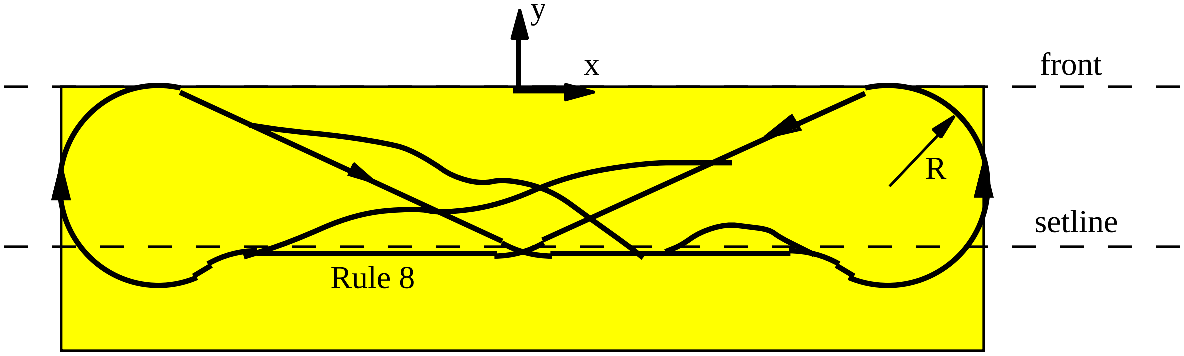
TACCSF (USAF)

Theater Air Command and Control Simulation Facility

- Human pilots and operators provide flight and fire control

A New Rule-based ABL Flight Controller

front rule:	IF (ahead of box or about to go ahead of box)	THEN turn backward
side rule a:	IF (past or passing side AND can complete forward turn)	THEN turn forward
side rule b:	IF (past or about to pass side)	THEN turn backward
retreat rule:	IF (flying backwards)	THEN turn forward
diagonal rule:	IF (ahead of diagonal back to last resort circle)	THEN turn backw
request rule:	IF (received a turn request)	THEN turn forward
setline rule:	IF (ahead of baseline)	THEN turn backward
default rule:	turn forward	



Typical orbit produced by this rule-based controller

Transform the Ruleset into an Adaptive Controller

Define new variables in terms of ruleset antecedents

v0: Is the plane past or about to pass a side of the box?
v1: Is the plane too close to the front to make a forward turn?
v2: Is the plane headed forward of the max heading?
v3: Can the plane reach the last resort turning circle?
v4: Is the plane in front of the setline?
v5: Is there a request for a forward turn?

c = 1, turn forward
c = 0, turn backward

Completely equivalent binary string representation of ruleset

```
v0    0101010101010101010101010101010101010101010101010101010101010101
v1    0011001100110011001100110011001100110011001100110011001100110011
v2    0000111100001111000011110000111100001111000011110000111100001111
v3    0000000011111111000000001111111100000000111111110000000011111111
v4    0000000000000000111111111111111100000000000000001111111111111111
v5    0000000000000000000000000000000011111111111111111111111111111111
      ----------------------------------------------------------------
c     1110111011100100111001001110010011101110111001001110111011100100
```

Highlighted example:

IF well within sides AND have room for forward turn AND heading rearward
AND behind the diagonal back to the last resort turning circle
AND in front of setline AND no turn is requested THEN turn forward

Each of the $2^{64} = 1.8\ 10^{19}$ possible strings represents an alternative rule-based controller

Evolutionary Search for Improved Controllers

the 64 bit string forms the chromosome

- Construct initial population by mutation from original ruleset string
- Evaluate the performance of each controller in the simulation
- Construct next generation
  - Performance based selection,cross-over, and mutation
- Evaluate next generation and sort population by performance
- Iterate more generations
  - mutation rate annealing

Evolutionary Training Finds Better Controllers

- Training set of 25 sorties of 110 missiles each (2750 engagements)

- Baseline ruleset performance: 1468 of 2750 killed (53.4%)

- Genetic search for improved controller: 117 performance evaluations
  New controller killed 1522 of 2750 (55.3%)

- The new controller was then tested on 11,000 independent engagements,
  Performance improved to 54.7% from the 53.8% of the original ruleset

The Evolved String Gives An Improved Ruleset

The flipped bits of the evolved controller transfer directly into new rules

Three new rules capture the new knowlege in the evolved controller

Example:
IF no turn is requested AND have room for forward turn AND heading too far rearward
AND behind the diagonal back to the last resort turning circle
AND well within sides AND in front of setline THEN turn backward

This controller has discovered new circumstances when it is better to start its forward turn a little early or to make a rearwards turn

Rule-based Fire Controller

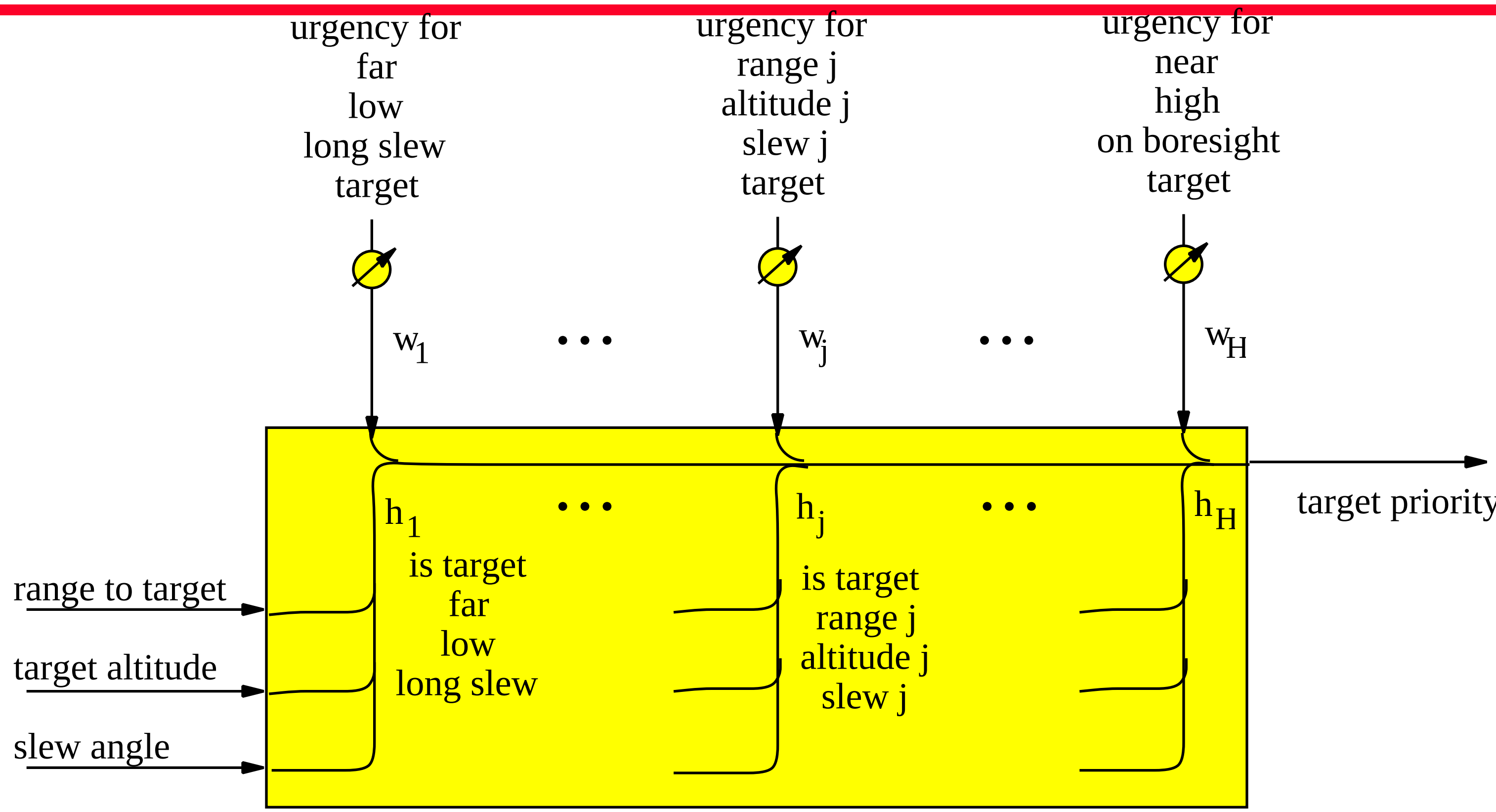
Shoot the engagable target with shortest estimated time to kill

$$\text{Time-to-kill} = \text{slew time} + \frac{\text{lethal fluence}}{\text{deliverable intensity}}$$

target priority depends strongly on three parameters:

- range to target
- target altitude
- slew angle

Adaptive Structure Representation of Controller



Transform the controller into an adaptive structure representation with a linear expansion on a set of functions in the input space

- Multilinear Expansion - h's are from multi-linear interpolation between surrounding centers
- Fuzzy Set Interpretation - the h's are membership functions
- Radial Basis Function Expansion - the h's are from radial functions centered at input space locations

Training Methods to Find Better Weight Values

Supervised Training to Learn a Ruleset

Generate a training set (~1000 input-output pairs) from the ruleset
Find the set of weights that best mimics the training set (least square error)

Gradient descent with singly presented training pairs (~backprop)
Batch mode gradient descent (more efficient option for linear nets)
Multilinear net can use efficient matrix inversion methods

Production Training to Improve Performance

Find improved weights through simulation against an ensemble of scenarios

Pivot and Random offset with simulated annealing
Downhill simplex
Genetic Algorithm

~100 evaluations required for improved 8 parameter controller
~1000 evaluations required for improved 27 parameter controller

ABL Adaptive Fire Control Proof-of-Principle

Simulate performance in example scenario

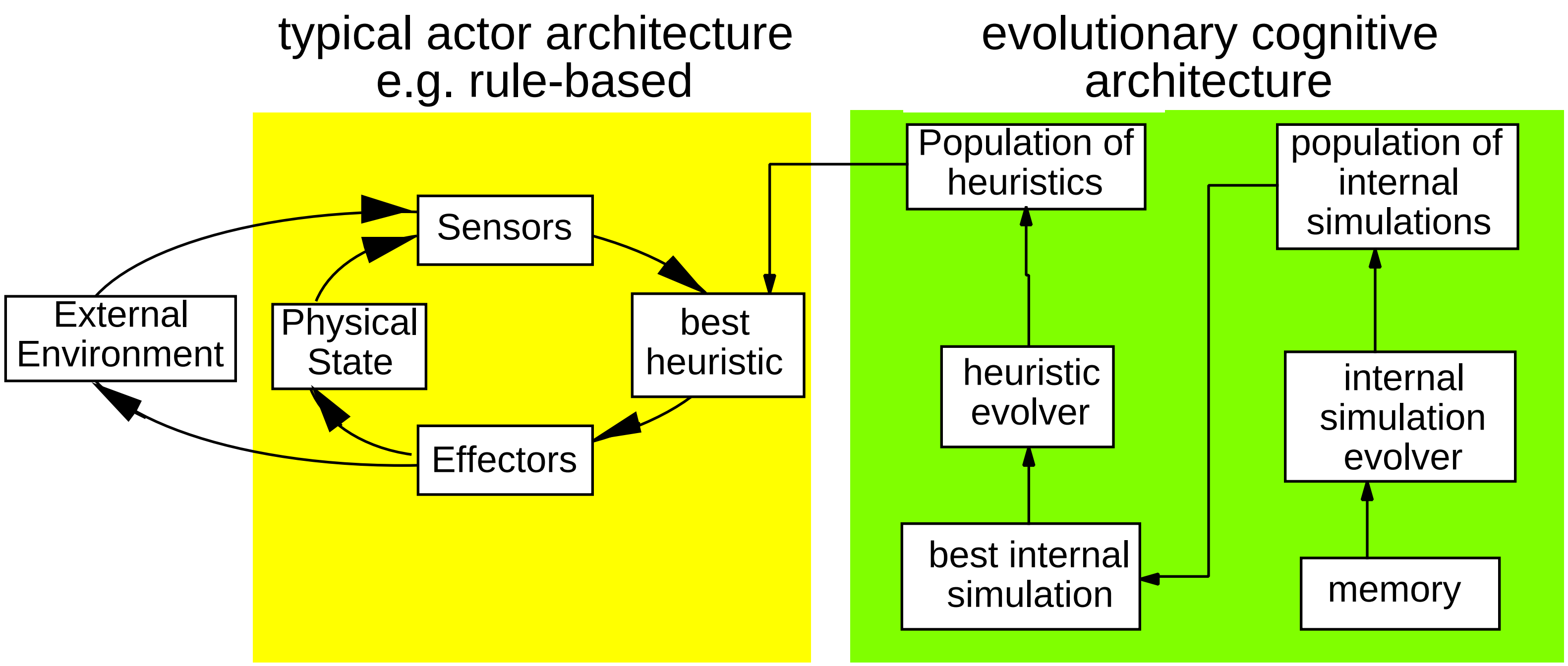
100 missiles, 600 km class,
launched within 200 seconds
launched from 300 by 200 km zone, 100 km behind FEBA
ABL stationed in 200 by 100 box, 100 km standoff

Baseline performance with shortest TTK target selection is **69** of 100 killed.

Train linear net to match shortest TTK target selection.
Net weights have direct interpretation as rule set.
Initial performance of linear net is **68** of 100 killed.

Allow 8 parameter linear net to evolve by pivot and offset using simulation based performance evaluation.
Performance of evolved linear net is **76** of 100 killed.
Evolved net weights have direct interpretation as new rule set.

Cognitive Architecture for On-line Learning



The Evolutionary Controller Improvement mechanism can be put inside the actor

- provides on-line learning capability
- actor evolves a population of behavior heuristics using an internal simulation
- internal population of simulations is evolved based on correspondence to real external environment

Concluding Remarks

- Methodology for automatic controller improvement
  - transformation of knowledge-based controller to an adaptable structure
  - controller performance evaluated through simulation
  - explore alternatives with efficient genetic algorithms.

- Can discover new behaviors, rules, tactics
  - Demonstrated dramatic improvements for two very different controllers

- Can adapt behaviors to alternative scenarios or different environments

- ABL controllers are just tip of iceberg
  - simulation actors for tank crew and J-Star ground station module operators
  - applications for military, economic, political, industrial, social systems

- Extension to evolutionary cognitive architecture for on-line adaptive ability